

<p>(51) International Patent Classification 7 : G06F 9/445, 17/30</p>	<p>A1</p>	<p>(11) International Publication Number: WO 00/45259</p> <p>(43) International Publication Date: 3 August 2000 (03.08.00)</p>
<p>(21) International Application Number: PCT/US00/01995</p> <p>(22) International Filing Date: 28 January 2000 (28.01.00)</p> <p>(30) Priority Data: 09/240,229 29 January 1999 (29.01.99) US</p> <p>(71) Applicant: ORACLE CORPORATION [US/US]; 500 Oracle Parkway, Redwood Shores, CA 94065 (US).</p> <p>(72) Inventors: ELLISON, Lawrence, J.; 500 Oracle Parkway, Redwood Shores, CA 94065 (US). DOHERTY, C., Gregory; 36 Downey Street, San Francisco, CA 94117 (US). ROSSITER, Jay; 164 Lois Lane, Palo Alto, CA 94303 (US). STOWELL, David; 18253 Chelmsford Drive, Cupertino, CA 95014 (US). NAKOS, Dimitris; 2220 Middlefield Road, Palo Alto, CA 94301 (US). BHADURI, Pari; 1043 A Foster City Boulevard, Foster City, CA 94403 (US). MALLA, Venkat; 6133 Linford Terrace, Fremont, CA 94555 (US). CHEN, Jane; 395 Richmond Drive #18, Millbrae, CA 94030 (US). KUMARESAN, Bala; 625 Old Country Road #215, Belmont, CA 94002 (US). JOHN, E., So, John; 1235 Jefferson Avenue #102, Redwood City, CA 94062 (US). BANERJEE, Niloy; 2675 Fayette Drive #207, Mountain View, CA 94040 (US). MATURI, Srinivas; 471 Hillsdale Avenue, Santa Clara, CA 95051 (US).</p>		<p>GANTI, Lalitha; 639 Old County Road #13, Belmont, CA 94402 (US). VENKATARAMAN, Ramkumar; 1208 Admiralty Lane, Foster City, CA 94404 (US). STHANIKAM Balasubramanyam; 1107 Second Avenue #501, Redwood City, CA 94063 (US). OU, Yuanjiang; 1230 Whipple Avenue #307, Redwood City, CA 94062 (US). KOONEN, Joseph, Therrattil; #754 9 Cross, Indira Nagar II Stage, Bangalore 560 038, Karnataka (IN). RAWAT, Harish; House No. 016, Mukund Apartments, Palmgrove Road, Bangalore, Karnataka 56004-7 (IN). OLAGAPPAN, Muthu; No. 9, Jeevaanandam Street, Vinayagapuram, Ambattur, Chennai, TamilNadu (IN). SIVAKUMAR, Thivakaram, Prakash; 7th floor Sona Towers, 71 Miller Road, Bangalore, 506 052 (IN).</p> <p>(74) Agents: HICKMAN, Brian, D. et al.; McDermott, Will & Emery, 600 13th Street, N.W., Washington, DC 20005-3096 (US).</p> <p>(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p>
<p>Published</p> <p><i>With international search report.</i></p> <p><i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>		

Diagram 100 illustrates a network architecture. A local network 102 is connected to a wide area network 150. The local network 102 includes a router 104, a firewall 100, a directory server 134, an integrated management console 130, and a backup server 132. The wide area network 150 is represented by a cloud. The database appliance 110 is connected to the local network 102 and contains a database server 112, operating system 114, hardware 116, and a database 122.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

PRE-DEFINED HARDWARE AND SOFTWARE BUNDLE READY
FOR DATABASE APPLICATIONS

FIELD OF THE INVENTION

The present invention relates to database systems and, more particularly, to reducing the total cost associated with maintaining one or more database systems.

BACKGROUND OF THE INVENTION

5 Inevitably, companies that invest in computer systems discover that the cost of owning a computer system is far greater than the price of the initial hardware and software purchases. Even relatively small computer systems, consisting of a handful of workstations networked together, frequently encounter problems that can consume an exorbitant amount of time and money. To prevent such problems, and to address
10 such problems when they occur, many companies hire full time information service (IS) professionals. Frequently, the size of a company's in-house IS department grows until it consumes a disproportionate amount of the company's resources and profits.

 The magnitude of the expenses associated with maintaining a computer
15 system is due in part to the fact that most computer systems are composed of numerous layers of general purpose components, even though the actual use of all layers is directed to a single specific application. For example, a company may purchase a computer system for the sole purpose of handling the company's accounting. However, the actual workstations purchased for this purpose will
20 typically be general purpose machines that provide all of the hardware services required to be able to handle any application. Many of these services are completely irrelevant for the actual use to which the machine will be directed. For example, the machines may come equipped with built-in "three-dimensional" sound capabilities that will never be utilized while the machines are used for accounting.

25 The general purpose machines that are to implement the accounting system will typically come installed with a general purpose operating system. Similar to general purpose hardware, general purpose operating systems attempt to provide

services to address all possible needs of all possible types of software. For example, general purpose operating systems employ complex I/O techniques for handling I/O intensive applications, complex techniques to support computation intensive applications, and complex communication techniques to support communication intensive applications. Many of those services may not be required by the specific context, such as accounting, in which the operating system may actually be used.

Typically, the hardware and operating system would not be the only "general purpose" components that would be used to implement an accounting system. For example, the accounting system may be configured to interact with a database system. Typically, the database system would provide services to address all possible uses of the database system, and not just those required by the accounting program. Given the diverse environments in which database systems may be used, the services actually required by any given application may be a small fraction of all the services provided by the database system.

In the accounting example given above, each of the general purpose components tends to be complex and, consequentially, difficult to manage. When many of such components are thrown together in a single system, the complexity increases exponentially, thus necessitating the significant after-the-purchase computer system maintenance expenses that most companies have encountered.

In the context of database systems, the complexity of the general purpose database system combined with the complexity of the general purpose operating system with which the database system interacts is typically sufficient to require employment of a full-time database administrator (DBA). It is the responsibility of the DBA to grapple with the complexity of the database system, the operating system, and the interactions between them, so that the other database users can take advantage of the database system without being exposed to underlying complexity. Unfortunately, employing a full-time DBA makes ownership of a useful database system an expensive option.

Based on the foregoing, it is clearly desirable to provide a computer system that employs features that reduce the cost of maintaining the database system without

having a negative impact on the ability of the computer system to perform its intended purpose. It is further desirable to decrease the complexity of computer systems, and to provide management techniques that allow companies to reduce the size and cost of their in-house IS departments.

SUMMARY OF THE INVENTION

A database system that incorporates numerous features that reduce the total cost of maintaining the database system is provided. That database system includes a database appliance that executes a database server on a platform that includes a
5 special purpose operating system specifically tailored to the services required by the database server. According to certain aspects, the hardware may also be specially tailored to the services required by the database server. According to one aspect of the invention, the database appliance configures itself by detecting the environment in which it resides and setting operational parameters based on the detected
10 environment. The configuration metadata of all components of the system are stored in a centralized repository which itself may reside external to the system.

According to another aspect, both the database server configuration and the operating system configuration are managed by a remotely located integrated management console, which interacts with and configures the system at the database
15 system level, the operating system level and, according to one embodiment, at the hardware subsystem level. Backup management may also be performed remotely. The remote components, such as the integrated management console, the backup server, and the configuration repository, may communicate with the system through a local area network or wide area network (e.g. the Internet), including through a dial-
20 up connection.

Using the features described herein, numerous benefits are achieved. Specifically, the ease of using a database system is increased by making the database system easier to set up, manage, and service. The total cost of ownership is reduced by avoiding the costs associated with operating systems and middleware (e.g. web
25 server), by providing remote management, by reducing down time, and by protecting the user's investment. Better performance is achieved through the use of a slim lined operating system, by tuning the database system for a dedicated application, and through the use of custom hardware. The availability of the database system is increased by avoiding problems associated with direct user access to the operating
30 system, and through the use of RAID and/or network attached storage.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

5 FIG. 1A is a block diagram that shows a system that implements an embodiment of the present invention;

FIG. 1B is a block diagram that shows alternative configuration for a system that implements an embodiment of the present invention;

10 FIG. 1C is a block diagram that shows alternative configuration for a system that implements an embodiment of the present invention;

FIG. 1D is a block diagram that shows alternative configuration for a system that implements an embodiment of the present invention; and

FIG. 2 is a block diagram of a machine that may be used to implement a database appliance according to an embodiment of the invention;

15 FIG. 3 is a flowchart that illustrates the steps for recovering a database appliance according to an embodiment of the invention; and

FIGS. 4A, 4B, 4C, and 4D collectively depict a flowchart that illustrates the steps for installing a database appliance software bundle on a machine to create a database appliance according to an embodiment of the invention.

20

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Techniques are described herein for providing a database system with reduced exposed complexity, reduced maintenance costs, and simplified management are described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

FUNCTIONAL OVERVIEW

A system that incorporates numerous features that reduce the total cost of maintaining the database system is provided. The system includes, among other things, one or more "database appliances". A database appliance is a machine that includes database server software and non-database server software, where the non-database software is specially tailored to the needs of the database server. Such non-database software may include, for example, operating system software from which have been removed services that are typically used in general purpose computers but are not required to support the needs of the database server. According to one embodiment, even the hardware of the machine deviates from a general purpose computer in order to more efficiently support specific characteristics and requirements of the database server.

The total cost of database system maintenance is further reduced by providing an integrated management console which, according to one embodiment, is located outside the local network in which the database system resides. The integrated management console manages all functional components (including hardware and software) of the database appliances. The maintenance costs of the database system are further reduced by having a single "outside consultant" administrator use the integrated management console to manage the database appliances of numerous

companies, thus avoiding the need for those companies to each employ their own expert database administrator.

EXEMPLARY SYSTEM

5 Referring to Figure 1A, it is a block diagram of a system 100 configured according to an embodiment of the present invention. System 100 includes local area network 102 that is connected to a wide area network 150 through a firewall 106. In a typical configuration, local area network 102 is a network installed at and owned by a particular company, and wide area network 150 is the Internet.

10 Within local area network 102, a database appliance 110 is connected to a set of storage devices 122 and to one or more database clients. Storage devices 122 are shown external to database appliance 110, but may alternatively be integrated in database appliance 110 itself. As used herein, the term "client" refers to a system or component that may reside either outside of appliance 110 (e.g. on workstation 108
15 in Figure 1A) or within the appliance 110. In the context of database systems, the client of a database server is typically referred to as a database application.

The local network 102 includes a router 104 that routes messages between remote devices on the wide area network 150 and the various components on the local network 102. In particular, system 100 includes an integrated management
20 console 130, a directory server 134 and a backup server 132, all of which can communicate with each other and with the components of local network 102 through wide area network 150.

System 100 incorporates numerous features that reduce the cost of maintaining local network 102 for the company that owns local network 102. Each
25 of the features shall now be described in greater detail.

The arrangement and content of system 100 is merely exemplary. The specific arrangement and content of a system that implements the features of the present invention will vary from implementation to implementation, depending on the specific needs of the user. Figures 1B-1D, for example, illustrate alternative
30 configurations. The system 140 shown in Figure 1B has the backup server 132,

directory server 134, and integrated management console 130 all within the local network. The system 142 of Figure 1C includes numerous database appliances. The system 144 of Figure 1D includes numerous local networks 170 and 172 that can be managed through the same integrated management console 130 as is used to manage
5 local network 102.

DATABASE APPLIANCE

Database appliance 110 is a machine configured to execute a database server 112. Similar to conventional database implementations, database appliance 110
10 includes a database server 112 and operating system 114 executing on hardware 116. However, unlike conventional database implementations, the operating system 114 executing on database appliance 110 is a special purpose operating system tailored specifically for executing database server 112. As a special purpose operating system, operating system 114 does not possess the high number and complexity of
15 services that general purpose operating systems employ. The nature and techniques for generating operating system 114 shall be discussed in greater detail hereafter.

According to one embodiment of the invention, hardware 116 is also specifically tailored for executing database server 112. The cache hit ratio experienced by the database appliance 110 may be improved by a special
20 configuration of the hardware. Specifically, at the hardware level, the operating system level, the database server level, and possibly at the database application level, data is moved between the levels by copying the data from one place in memory to another. In a general purpose computer system, each layer will typically use different buffers for storing and copying the data, and different algorithms for
25 determining which data should be stored in its buffers. In contrast, according to one embodiment of the invention, from the database server level to the level of the microchips in hardware 116, the same algorithm is used for determining which data is to be cached.

GENERATING A SPECIAL PURPOSE OPERATING SYSTEM

Various techniques may be employed to generate an operating system 114 specifically tailored to execute database server 112. In general, the generation of operating system 114 may involve initially writing a special purpose operating system based on the needs of database server 112, or modifying an existing general purpose operating system to create operating system 114. In either case, a first step in the generation of operating system 114 is determining the services needed by database server 112.

According to one embodiment, the source code of database server 112 is examined to identify calls made to operating system functions. The operating system calls thus identified are considered calls to "required services". After the required services are identified, operating system 114 is generated to provide specific support of those required services.

In an embodiment that generates operating system 114 by modifying an existing general purpose operating system, operating system 114 may be generated by (1) removing operating system code that corresponds to services that are not required services, (2) tuning the parameters and configuration of required services to achieve better performance, and (3) modifying the code that implements required services to enhance their efficiency relative to their use by database server 112.

According to one embodiment of the invention, operating system 114 does not provide services directly to end users. For example, in an embodiment where operating system 114 is generated by modifying an existing general purpose operating system, one of the modifications made to generate operating system 114 involves removing from the general purpose operating system all routines that provide an interface through which an end user can directly access the operating system. Thus, an end user may cause changes to be made to the file system on disks 122 in response to interaction with database server 112, but not through direct calls to the file system services of operating system 114.

In many operating systems, operating system services are provided by both service modules and by an operating system "kernel", where the service modules

execute in their own address space separate from the kernel. The kernel of operating systems that rely heavily on user-space modules to provide services is commonly referred to as a microkernel. Special purpose operating system 114 may be generated by modifying a general purpose operating system that employs a single monolithic
5 kernel, or by modifying a general purpose operating system that employs a microkernel and associated service module.

The following techniques apply to both operating systems that do and do not employ a microkernel. Modifying the general purpose operating system may involve performing the above-listed steps of removing extraneous code, tuning parameters,
10 and enhancing code at both the service module level and at the kernel level. In some cases, entire service modules may be removed when the services they provide are not required by the database server 112. For example, a general purpose UNIX operating system may have the following service modules:

NFS Daemon – Network File System I/O Server.
15 Printer Daemon – Print spooling service provider.
FTP Daemon – File Transfer Protocol Server, provider of file transfer protocol services.

INET Daemon – A listener listening in specific Internet sockets for connections. Based on requests, the INET Daemon starts services to service the
20 requests. For example, the INET Daemon listens for “bootp” requests and starts a “bootp daemon” upon receiving a bootp request.

Time Daemon – Time keeper to synchronize host’s time with other machines in the network.

HTTP Daemon – A web server.
25 PPP Daemon – Dial up daemon that provides dial-up support.
Mount Daemon – The Network File System Mount Request Server.
Swapper Daemon – A system process that is invoked when free memory falls below a minimum. The swapper daemon scans an “inactive_pagelist” and marks pages to be swapped out, based on some replacement technique.

Flush Daemon – A system process that flushes the dirty buffers of a buffer cache back to the file system.

Telnet Daemon – A service that provides an interactive session with the host to external clients.

5 Many of these service modules provide services that are not required by database server 112. Operating system 114 is generated, in part, by removing from the general purpose operating system those unused service modules. In the context of database appliance 110, the modules that are removed from the general purpose operating system to generate operating system 114 may include, for example: the
10 NFS Daemon, the FTP Daemon, the Mount Daemon, and the Telnet Daemon.

In embodiments that generate operating system 114 by modifying an existing general purpose operating system, the identification of the services to be optimized may alternatively or additionally be performed using a kernel trace. While database server 112 is being used, a code trace is performed by tracking the thread of
15 execution. When database server 112 requires services from the operating system, the database server code will call portions of operating system 114, causing those portions to be identified in the thread of execution. After the trace is performed, the services provided by those portions of the operating system that are identified in the trace are considered good candidates for optimization. For example, if the trace
20 shows that a particular operating system routine was invoke one hundred times more than other operating system routines that were invoked during the same time period, the frequently invoked routine may be selected as a candidate for optimization. The kernel trace may even provide hints with respect to how those portions should be optimized.

25

ENHANCING AND TUNING REQUIRED SERVICES

In addition to discarding operating services that are not required by a particular application, generating an application-tailored operating system may also involve enhancing those operating system services that are required. For example,
30 assume that an operating system is to be generated for an appliance that is to operate

as an Internet file server. Further assume that the file server is to store articles about the daily news. Due to the nature of the application, the appliance may be accessed through thousands of concurrent connections. Further, most of those connections will involve requests for the same material. A high number of buffers is required to support that number of connections, but once the heavily-accessed material is loaded into the buffers, very little disk access will be required.

Under these conditions, if the operating system is not specially tailored for such access, the operating system may allocate fewer buffers than are required to store the entire amount of heavily accessed material. Consequently, the buffers may be overwritten with such frequency that it becomes difficult for any user to interact efficiently with the appliance. A specially tailored Internet file server application, on the other hand, may include a database that dedicates relatively fewer resources to efficiently retrieve the material, and relatively more resources for storing the heavily accessed material once it has been retrieved. Consequently, buffered portions of the heavily accessed material will not have to be overwritten to load in other portions of the heavily accessed material, since enough buffers have been allocated to store the entire set of heavily access material. Because portions of the heavily accessed material are not being overwritten, those portions do not have to be retrieved again from disk when subsequently requested. Ironically, by reducing the resources used for disk retrieval, the efficiency of disk retrieval can actually be improved because fewer disk accesses will be required.

The shifting of resources from disk retrieval to cache storage is an example of an enhancement that may be performed by tuning the configuration and parameters of the services involved, as opposed to actually modifying the code of the underlying services. Such tuning techniques may be used instead of, or in addition to, actual code optimizations of critical required services. Actual code modifications may involve, for example, causing operating system 114 to store resource information in a way that allows it to be directly read by database server 112 without causing a context switch to the operating system kernel address space, and causing database

server 112 to read the resource information directly rather than by performing system calls through the system call interface of the operating system.

In the context of OLTP database applications, much of the workload of database appliance 110 will involve I/O. Specifically, the workload of database appliance 110 is largely retrieving information from tables stored on non-volatile storage devices 122, caching the information that is used frequently, and supplying that information to database application workstations 118 over the local network 102. In general, the actual computational demands on database appliance 110 will be minimal, such as calculating the balance of a bank account. Therefore, when the anticipated application of database appliance 110 is as a server to OLTP applications, the process of enhancing certain required services would typically involve enhancing or dedicating relatively more resources to the operating system services responsible for disk and network I/O. In many commercially available operating systems, the services that handle disk I/O are referred to as Async I/O, Direct I/O, raw device access, and striping.

One technique for optimizing required services involves, for selected operations, bypassing certain layers that would normally be involved in those operations in a general purpose machine. Typically, such layers are provided in general purpose operating systems, for example, to give a virtual common interface. Though convenient, the overhead required to provide a virtual common interface negatively impacts performance. The database appliance may be optimized to bypass unwanted layers, thereby reducing overhead.

In trying to solve all problems for all situations, general purpose components frequently do not provide the optimal solution for a particular problem. For example, in a general purpose operating system, the allocation of resources (e.g. the CPU and memory) is performed so that all consumers are given a fair share of the resource. However, within a database appliance designed according to an embodiment of the invention, the processes that will be executing are generally known. Therefore, one technique for optimizing required services involves designing the operating system to allocate resources to the known processes in a manner that best improves

performance of the appliance. The conditions and demands of various processes do not remain the same throughout the life of the processes. Therefore, according to one embodiment, an adaptive resource manager is provided to get the appliance to perform optimally, even when the load conditions and demands change.

5 Allocating a certain percentage of CPU time to a process is different than allocating to the process CPU time when the process needs it. According to one embodiment, the scheduling techniques of the special purpose operating system are based on shares of CPU time, as allotted by the adaptive resource manager. The adaptive resource manager allots CPU shares and process priority based on the
10 changing resource demands of the process. The parameters used by the adaptive resource manager can be set by an administrator, or be automatically determined based on historical information maintained about the processes.

DATABASE APPLICATION SPECIFIC DATABASE APPLIANCES

15 In addition to tailoring the operating system 114 and the hardware 116 of a database appliance 110 to a particular database server 112, database server 112 itself along with the operating system 114 and the hardware 116 may be more particularly tailored to a particular database application. For example, one type of database application, referred to as an online transaction processing (OLTP) application, is
20 relatively I/O intensive, while another type of database application, referred to as an online analytical processing (OLAP) application, is relatively computationally intensive. Consequently, at each level, the database appliance configured for use with an OLTP application is adapted to enhance I/O performance of database server 112, while the database appliance configured for use with an OLAP application is
25 adapted to enhance computation performance.

SELF-CONFIGURING DATABASE SYSTEM

When a company purchases a new system, such as local network 102, the company typically pays a third party to "install" the system. The process of
30 installing even smaller systems may involve a significant expenditure of time and

money, in part because each of the complex general purpose of the components need to be configured to interact accurately with each of the other complex general purpose components on the system.

According to an embodiment of the invention, database appliance 110
5 includes self-configuration software that avoids the need to hire installation experts. When executed, the self-configuration software detects the environment in which database appliance 110 is being used, and configures database appliance 110 according to the detected environment. For example, the same bundle of database appliance software is sold to a first company that will have a database cache of 200
10 megabytes and have a 10 gigabyte database, and to a second company that will have a database cache of 32 megabytes and a 500 megabyte database. The first company executes the database appliance software in a system that includes significantly more resources than the second company. In response to detecting the resources available in the first company's system, the database appliance configures itself to support a
15 200 megabyte database cache and a 10 gigabyte database. Similarly, upon detecting the resources available at the second company, the database appliance configures itself to support a 32 megabyte database cache and a 500 megabyte database.

According to one embodiment, the self-configuration includes adjusting parameters that affect both the configuration and operation of database server 112,
20 and the configuration and operation of operating system 114. Optionally, the self-configuration software further adjust parameters that affect the operation of the hardware 116 used by the database appliance 110.

Figures 4A-4D collectively depict a flowchart illustrating steps for installing the bundle of appliance software on a machine to create a database appliance.

25 According to one embodiment, the installation process is a "1-button" process in that it completes all operations without needing any human intervention during the process. Further, the appliance software bundle does not simply install on the machine, but rather it configures the system in such a way that when booted up, the system starts up all needed services, and is ready to be used to run database
30 applications. Additionally, the installation process detects what hardware is present,

and auto configures to allow it to be entirely dedicated to run the database and related programs only. If the system has to be upgraded or recovered from a system disk failure, the same 1-button process is used to upgrade or recover the appliance.

Referring to Figures 4A-4D, in step 400 the appliance is booted from a
5 removable storage medium, such as a CD ROM or floppy disk, the CD ROM is mounted and a memory file system (MFS) is created. The MFS is a file system created in the volatile memory of the machine. It is used as a temporary work area until the non-volatile disks are formatted. In step 402, a search is performed for the devices that are required by the software being installed. In step 404, it is determined
10 whether any required device is absent. If any required device is absent, control flows from step 404 to step 406 and the installation operation is terminated.

If all of the required devices are present, control passes from step 404 to step 408. At step 408, it is determined whether the Dynamic Host Control Protocol Daemon (DHCPD) specifies any site specific information. In particular, a DHCP
15 client broadcasts packets for a DHCP daemon on the same subnet. The DHCPD then responds with network parameters, as well as other site-specific parameters. If site specific information exists, control passes from step 408 to step 12. Otherwise, control passes from step 408 to step 410.

At step 410, it is determined whether there is any site specific information
20 specified on the installation floppy disk. If the installation floppy disk does contain site specific information, control passes to step 412. Otherwise, control passes to step 416.

At step 412, a SITE-SPECIFIC flag is marked as "done". Control flows from step 412 to step 414, where the site specific information is parsed. Control
25 passes from step 414 to step 416.

At step 416, it is determined whether an installation already exists on the appliance. If an installation already exists, control passes to step 424. Otherwise, control passes to step 418. At step 424, the setup procedures for an upgrade operation are performed, and control passes to step 428. If there is no existing
30 installation, at step 418, it is determined whether recovery is being performed from a

CD ROM. If recovery is being performed, control passes to step 420. Otherwise, control passes to step 426.

At step 420, recovery of distribution is performed from the recovery CD ROM. Specifically, when the system disk fails, the appliance cannot boot up unless
5 a new disk is substituted for the failed disk, and the distribution and data on the failed disk are reconstructed. Database data on the system disk is backed up to its pair disk. However, the distribution is obtained from the install CD ROM. The system boots from the CD ROM, automatically determines that it is involved in a recovery from the CD, formats the new disk, recovers the distribution from the CD, and recovers
10 data from backup. Control passes from step 420 to step 422 where the installation process is terminated.

At step 426, the storage devices of the appliance are partitioned and labeled, and file systems are created thereon. At this point, the installation software ceases to use the MFS for temporary storage, and begins to use portions of the appliance's
15 persistent storage or temporary storage. Control passes from step 426 to step 428.

At step 428, a windows manager (e.g. X) is started. If no display device is present on the appliance, then the windows manager is started in "quiet mode" using a virtual frame buffer. Control passes from step 428 to step 430. In step 430, the environment for the installation software is set, and the installation software is
20 started. Control then flows from step 430 to step 432.

At step 432, it is determined whether the SITE-SPECIFIC flag is set to "done". If the flag is set to done, then control passes to step 436. Otherwise control passes to step 434. At step 434, install scripts that are related to the site-specific parameters are copied to the persistent storage of the appliance. These install scripts
25 are preferable run when the appliance is rebooted when the site-specific parameters are available. Control then passes to step 436.

At step 436, the install software installs the operating system and site specific configuration parameters for the operating system. Control then passes to step 438 where the installation software installs the database server along with the site specific
30 configuration parameters for the database server. Control then passes to step 440.

At step 440, the installation software installs web server software along with site specific configuration parameters for the web server. Control then passes to step 442.

At step 442, the install software installs web database software and site specific configuration parameters for the web database software. The web database software is a software module that translates web requests into database requests. For example, the web requests may arrive according to HTTP protocol, and be converted by the web database software to conform to the database language used by the database server employed by the appliance (e.g. SQL or PL/SQL). Control then passes to step 444. At step 444, backup, recovery, and scaling boot scripts are generated. These scripts enable automatic backup or recovery, as well as providing means to expand the system when additional hardware is included. Control then passes to step 446, where other boot/cron/admin scripts are generated. The boot scripts dictate the steps performed on boot up to get the system up and running without manual intervention. The cron scripts run periodic jobs to do system management tasks. For example, they may be configured to automatically run processes for a remote administrator in the even of a crash. The admin scripts perform management tasks in response to events and/or commands from the remote console. Control then passes to step 448 where an initial backup is performed, and at step 450 the installation process is terminated.

INTEGRATED MANAGEMENT CONSOLE

Integrated management console 130 is one of the many feature of system 100 that reduce the total cost of ownership of the database system implemented on the local network 102. Integrated management console 130 deviates significantly from prior approaches to system management in that it integrates management of the database appliance 110 at numerous levels, including two or more of the database application level, the database server level, the operating system level, and the hardware level.

According to the prior approaches, each of those levels represented a separate general-purpose product, where each of the general purpose products is typically provided by a different source. Each of the different sources would provide separate management software for its own general purpose product. Consequently, systems
5 would typically include one console running management software from one source for managing a database server, another console running management software from another source for managing an operating system, and yet another console running management software from another source for managing the hardware. Typically, each management package would itself be complex, due to the inevitable complexity
10 of the general purpose product that they are intended to manage. Further, because the management console for each component was provided by a different source, the consoles would not share a common user interface or management model, thereby significantly increasing the cost of management.

The integrated management console 130 is a console through which
15 numerous aspects of the database system on local area network 102 can be remotely managed. Specifically, the company that owns local area network 102 need not have its own in-house department of database and system administrators. Rather, a third party consultant is able to efficiently manage the company's database system remotely. In fact, the same third party consultant is able to manage the database
20 systems of numerous companies in the same manner and through the same integrated management console. Consequently, the cost of the consultant's work can be shared among the many companies.

Integrated management console 130 allows an administrator to remotely manage database server appliances, both at the database server level and at the
25 operating system level. In embodiments where the database application itself is also located on the database appliance 110, the database application is also managed through the integrated management console 130.

Optionally, integrated management console 130 is also used to manage database appliance 110 at the hardware level. For example, hardware 116 may
30 include a redundant array of inexpensive disks (RAID) subsystem whose operational

parameters and configuration can be managed through integrated management console 130.

Because the numerous aspects of the database appliance 110 are managed through the same console that uses a consistent user interface and management model, the management chore is simplified. Further, management of the whole
5 database appliance is significantly less burdensome than management of each of the parts of previous database systems, because each of the parts managed by integrated management console 130 is specifically tailored to the particular context in which the appliance will be used, and lacks the innate complexity of its general purpose
10 counterpart.

According to one embodiment, database appliance 110 includes a simple network management protocol (SNMP) compliant agent and a multi-level management agent. The integrated management console interacts with the SNMP agent using the SNMP protocol. The SNMP agent, in turn, is configured to send
15 commands to the multi-level management agent in response to the messages received from the integrated management console. The multi-level management agent responds to the commands by performing management functions at the various levels of database appliance 110.

In the embodiment illustrated in Figure 1, the integrated management console
20 130 connects to and communicates with database appliance 110 through a wide area network 150, such as the Internet. However, according to an alternative embodiment, integrated management console 130 is configured to connect to database appliance 110 through a direct dial-up connection, thus bypassing the need for infrastructure required to connect database appliance 110 to the wide area network 150, and the
25 additional infrastructure required at the other end to connect integrated management console 130 to the wide area network 150. In some embodiments, the direct dial-up connection would still pass through a company's firewall for security reasons.

According to one embodiment of the invention, the integrated management console 130, along with the agent running on the appliance, allows administrators to
30 perform the following operations remotely:

- configure and control the operating system and database services,
- apply patches and upgrade to a set of appliance servers,
- enable lights-out management, and
- generate reports of the system resource usage and optimize performance.

5 With respect to lights-out management, a remote administrator may use the integrated management console 130 to define a set of "events" that may represent some error conditions. The administrator can then associate with each event a "fixit" job to be invoked when the event is triggered. The administrator can also
10 configure the system to raise an alert and send page or e-mails to the administrator on duty in response to certain types of critical alerts. For example, when the disk space containing archive logs is getting full to a definable threshold, an event can be triggered. The event may invoke a job that does a back up of the database and cleans up the archive logs. In response to other events, such as when a node or the network goes down, the system can call attention of the remote administrator by paging the
15 administrator.

BACKUP SERVER

Backup server 132 is a server configured to back up critical information that is stored in local network 102. Backup server 132 includes a tape storage, or some
20 other relatively inexpensive high capacity storage medium, for storing the data. The data from local network 102 to be backed up by backup server 132 is communicated to backup server 132 according to a backup schedule that will vary from implementation to implementation. Various mechanisms may be used to convey the data from the devices on which it resides within local network 102 to the remotely
25 located backup server 132, which does not reside on local network 102. According to one embodiment, the data is communicated through the firewall 106 over the wide area network 150 to be backed up on backup server 132. Alternatively, backup server 132 can connect to local network 102 through a non-dedicated connection.

Backup service commands that dictate operational parameters for backup
30 server 132 are, according to one embodiment, issued and controlled through the

integrated management console 120. Such commands determine the schedule by which backup server 132 backs up the data of local network 102. For example, integrated management console 130 may program backup server 132 to perform incremental backups nightly, and full backups on a weekly basis.

5 Because the backup of local network 102 is performed remotely, the company that owns local network 102 does not have to employ an on-site operator to perform the backups. Further, the operator that does perform the backups at the remote site may be a third party that performs such backups for numerous companies, thus spreading the cost among the numerous companies to reduce the cost required by
10 each of the individual companies.

DIRECTORY SERVER

A directory service is a repository for information (metadata) about a system. Communication with the directory service is performed through a protocol supported
15 by the directory service, such as the Lightweight Directory Access Protocol (LDAP). To find out information about a device, a search is performed on the directory service for the system. For example, a device outside of the system may wish to communicate with a device within the system. The device outside the system may only have the name of the device with which it desires to communicate. Using the
20 name, the directory service searches the metadata to determine the address of the named device. The address is returned to the outside device, which is then able to communicate with the inside device using the address.

According to one embodiment of the invention, all of the metadata about local network 102 is stored in a repository within directory server 134, which resides
25 outside of local network 102. The metadata includes, for example, information about how the system within local network 102 is configured, including the configuration of database appliance 110. In systems that include numerous database appliances, the metadata includes one set of configuration parameters that are common to all database appliances, and sets of configuration parameters that are specific to the
30 configuration of each of the individual database appliances. According to one

embodiment, a copy of the configuration data from the repository is also stored in a cache on a device within local network 102 to improve the speed at which it may be accessed by devices within local network 102.

Storing the configuration metadata for the components that reside within
5 network 102 in a centralized repository, and locating that repository outside the system itself, provides numerous benefits. One such benefit involves having a centralized point of control for tuning operations. For example, an operator of integrated management console 130 is able to change any of the configuration parameters of network 102 to "tune" the system for better performance by modifying
10 the metadata contained in directory server 134. Such configuration modifications are then propagated from the directory server 134 to the local network 102 to effect the changes.

Another benefit of the centralized repository involves replicating a tuned system. For example, assume that a company has hundreds of retail stores with
15 similar computing needs. One of the stores may have implemented a database system prior to the other stores. Over time the store has customized and tuned its configuration to perform optimally for its intended use. At some point in time, the company may desire to automate all of its stores. Because the one automated store has a central repository that stores its configuration information, that configuration
20 information can be easily inspected and copied to implement identically configured "pre-optimized" systems at the various other stores.

Another benefit involves the recovery from catastrophic failures, which shall be described in greater detail hereafter.

25 FAILURE RECOVERY

Subsystems within company's computer system typically provide their own mechanisms to facilitate recovery from a single point of failure, such as the failure of a disk or a network card. For example, RAID subsystems are designed to automatically correct single disk failures. When a subsystem fails, recovery is
30 slightly more complicated. For example, if an entire RAID subsystem fails, the data

stored thereon typically must be recovered from backups.

When a more catastrophic failure occurs, the entire computer system of a company ceases to function. Typically, such failures would also result in the loss of metadata that describes how the system was configured, such as the identity of the devices that belonged to the system, the network parameters that were in effect, the IP addresses that were being used, etc. In conventional computer systems, this information is not gathered in any particular location, but rather disbursed unsystematically among the components to which the configuration information applied. Consequently, recovering after such a failure was extremely difficult and expensive.

For example, if a system similar to the company's system exists, the company's system can be reconfigured based on the similar system, and the data can then be reloaded from backup. However, such a "similar system" may not exist if the company's system has been customized, or may exist but not be known to or controlled by those responsible for recovering the company's system. Further, some configuration information, such as the IP addresses used by the system, may be unique to the system.

In the case of a system failure, significant benefit results from having a centralized repository for the configuration metadata of local network 102, and even further benefit results from having that centralized repository located outside of the system to which it corresponds. Specifically, by maintaining a configuration repository for a computer system, such as local network 102, at a single location within the system, recovery from a catastrophic failure may be accomplished in two relatively inexpensive phases: (1) restore the configuration repository from backup to recover the configuration of the system, and (2) restore the data from backup onto the configured system. If the configuration repository is maintained outside the system itself, as in the embodiment illustrated in Figure 1, the first phase is even simpler because the repository itself would typically not have to be recovered from backup because it would not be lost in the catastrophic failure.

Figure 3 is a flowchart of a process for recovering a database appliance from

a crash according to an embodiment of the invention. Referring to Figure 3, at step 302 the integrated management console is used to search the metadata repository maintained by Directory Server 134 for the entry associated with the crashed appliance. According to one embodiment, the directory entry for the crashed appliance includes a unique appliance identifier for the appliance, data that identifies the class of the appliance, and data that indicates the version of the application software bundle that was installed on the crashed appliance.

At step 304, the same version of the appliance software that was installed on the crashed appliance is installed on a new appliance that is of the same class as the crashed appliance. The installation is performed using the appliance identifier of the crashed appliance, thus creating an association between the new appliance and the directory entry of the crashed appliance. At this point, the metadata repository has a valid directory entry associated with the appliance identifier of the crashed appliance, and the new appliance is set up to point to the crashed appliances entry in the metadata repository, even though the configuration of the new appliance is still the default configuration, not that of the crashed appliance.

At step 306, the integrated management console is used to set a flag, referred to herein as the "cache not in sync" flag, in the directory entry of the crashed appliance. At step 308, the new appliance is rebooted. At step 310, the new appliance checks its associated directory entry, which was the directory entry of the crashed appliance, and detects that the cache not in sync flag is set. In response to detecting this flag, at step 312 the new appliance performs auto-configuration by configuring itself to reflect the configuration information in its directory entry.

At this point, the integrated management console may be used to identify the backup solution used by the crashed appliance, and to initiate a restore operation to restore to the new appliance the information that was backed up from the crashed appliance (step 314). At step 316, any additional components or patches that had been installed on the crashed appliance are installed on the new appliance. The console administrator could initiate the applying of patches and install of additional components by browsing the crashed appliance's software component list in the

metadata repository.

HARDWARE OVERVIEW

Figure 2 is a block diagram that illustrates a computer system 200 upon which an embodiment of the invention may be implemented. Computer system 200 includes a CPU 204 for processing information. Computer system 200 also includes a memory 206, such as a random access memory (RAM) or other dynamic storage device, and a storage system 210, such as a magnetic disk or optical disk. Alternatively, storage system 210 may be located outside of computer system 200, and may be accessed by computer system 200 over the network via a standard network protocol, such as NFS. Additionally, EEPROM modules or variants, such as NVRAM, store instructions and data used to help bring the system up.

Other embodiments of the invention may use a computer system with built in controllers on the motherboard to enable tighter integration of the hardware components to achieve better performance. In such an embodiment, the system is scaled by clustering multiple integrated units.

The computer system can be expanded or scaled up with the addition of processors, memory modules, or storage units. When this occurs, the software adaptively configures itself to enable use of added components by the appliance.

Computer system 200 may be coupled to a display 212 for displaying information to a computer user. Display 212 may be, for example, a conventional computer monitor or merely a simple array of LEDs configured to convey status information. An input device 214 may be provided to allow direct input into computer system 200, in addition to input communicated through the network. Input device 214 may be a simple keypad, and/or may involve more sophisticated input mechanisms, such as a keyboard or mouse.

The invention is related to the use of computer system 200 as or in conjunction with a database appliance. According to one embodiment of the invention, a database appliance is implemented on computer system 200 in response to CPU 204 executing one or more sequences of one or more instructions contained

in memory 206, where the instructions include a database server and a special purpose operating system adapted to specifically provide the services of the database server and not all of the services required of a general purpose operating system.

Instructions may be read into memory 206 from another computer-readable medium, such as storage system 210. Execution of the sequences of instructions contained in memory 206 causes CPU 204 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to CPU 204 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 210. Volatile media includes dynamic memory, such as main memory 206. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 202. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

Computer system 200 also includes a network card 218. Network card 218 provides a two-way data communication coupling to a network link 220 that is connected to a local network 222. Network card 218 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

Network link 220 typically provides data communication through one or more networks to other data devices. For example, network link 220 may provide a connection through local network 222 to a host computer 224 or to data equipment operated by an Internet Service Provider (ISP) 226. ISP 226 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 228. Local network 222 and Internet

228 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 220 and through network card 218, which carry the digital data to and from computer system 200, are exemplary forms of carrier waves transporting the information.

5 Computer system 200 can send messages and receive data, including program code, through the network(s), network link 220 and network card 218. In the Internet example, a server 230 might transmit a requested code for an application program through Internet 228, ISP 226, local network 222 and network card 218.

 The received code may be executed by CPU 204 as it is received, and/or
10 stored in storage device 210, or other non-volatile storage for later execution. In this manner, computer system 200 may obtain application code in the form of a carrier wave.

 While various techniques for reducing the cost of maintaining a computer
15 system have been described above in the context of a database system, they need not be limited to any particular context. For example, rather than have a database appliance 110, a computer system may include a mail server appliance for handling electronic mail. Such a server would include, for example, e-mail server software, where the appliance levels below the server software (i.e. possibly a database system,
20 an operating system, and hardware) are specifically tailored to the needs of the e-mail server software.

 In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader
25 spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

- 1 1. A database appliance comprising:
2 a database server; and
3 a special purpose operating system whose features and configuration are
4 dictated by the database server and supporting components.
- 1 2. The database appliance of Claim 1 wherein the database appliance includes
2 self configuration software that automatically:
3 detects the environment in which the database server is installed; and
4 configures said database server based upon the detected environment.
- 1 3. The database appliance of Claim 1 wherein the database appliance includes
2 special purpose hardware specifically configured based on services required
3 by said database server.
- 1 4. The database appliance of Claim 1 wherein said database server is specially
2 configured based upon the services required by a specific database
3 application.
- 1 5. The database appliance of Claim 4 wherein said specific database application
2 is an online transaction processing application and said database appliance is
3 configured with relatively more resources dedicated to I/O services and
4 relatively fewer resources dedicated to computational services.
- 1 6. The database appliance of Claim 1 further comprising a management agent
2 component configured to change parameters associated with said database
3 server and said operating system in response to commands.

- 1 7. The database appliance of Claim 1 wherein said operating system includes a
2 kernel loaded in a particular address space, wherein said database server is
3 configured to read at least one item of information maintained by said
4 operating system by directly reading memory storing said item without
5 switching context to said particular address space of said kernel.
- 1 8. The database appliance of Claim 1 wherein the operating system is configured
2 to prevent direct end user access to services provided by said operating
3 system.
- 1 9. A method for implementing a database appliance, the method comprising the
2 steps of:
3 generating a special purpose operating system that is customized for
4 executing a database server; and
5 running said database server on a platform that includes said special purpose
6 operating system.
- 1 10. The method of Claim 9 wherein the step of generating said special purpose
2 operating system includes:
3 determining services required by said database server; and
4 writing an original operating system to specifically support those services
5 required by said database server.
- 1 11. The method of Claim 9 wherein the step of generating said special purpose
2 operating system includes:
3 determining services required by said database server; and
4 modifying a general purpose operating system to create said special purpose
5 operating system.

- 1 12. The method of Claim 11 wherein the step of determining services required by
2 said database server includes inspecting code of said database server to
3 identify operating system calls made by said database server.
- 1 13. The method of Claim 11 wherein the step of determining services required by
2 said database server includes tracing executing of said database server to
3 determine which portions of said general purpose operating system are used
4 during execution of said database server.
- 1 14. The method of Claim 11 wherein the step of modifying the general purpose
2 operating system is performed by removing from said general purpose
3 operating system all interfaces that allow direct end user access to services
4 provided by said general purpose operating system.
- 1 15. A method for managing a system that includes a plurality of devices arranged
2 in a network, the method comprising the steps of:
3 gathering and storing in a centralized repository metadata that reflects
4 configuration information about said system, and about each device of
5 said plurality of devices; and
6 managing configuration of said system based upon the metadata within said
7 centralized repository.
- 1 16. The method of Claim 15 further comprising performing the following steps in
2 response to a failure of the system:
3 recovering the centralized repository from a backup;
4 using the metadata within the centralized repository to configure the system;
5 after the system is configured, recovering the system.

- 1 17. The method of Claim 15 wherein the step of gathering and storing in a
2 centralized repository metadata is performed by gathering and storing in a
3 centralized repository metadata that resides outside said system.
- 1 18. The method of Claim 15 further comprising performing the following steps in
2 response to a failure of the system:
3 configuring the system based on the metadata restored in the centralized
4 repository; and
5 after the system is configured, recovering the system.
- 1 19. The method of Claim 15 further comprising the step of replicating said system
2 by performing the steps of:
3 copying said metadata to a second centralized repository associated with a
4 second system; and
5 configuring said second system based on the metadata contained in said
6 second centralized repository.
- 1 20. A method for managing a database system including one or more database
2 servers, the method comprising the steps of:
3 executing, at a console, an integrated management module configured to
4 manage at least two of the following layers in the database system: a
5 database application layer, a database server layer, an operating
6 system layer, and a hardware layer; and
7 interacting with a user interface provided by said integrated management
8 module to change operational parameters of said at least two layers.
- 1 21. The method of Claim 20 wherein:
2 the database server layer includes one or more database servers;
3 the operating system layer includes one or more operating systems that reside

4 on machines that are executing said database servers;
5 the integrated management module is configured to manage the one or more
6 database servers and one or more operating systems; and
7 the step of interacting with the user interface provided by said integrated
8 management module changes operational parameters of said one or
9 more database servers and said one or more operating systems.

1 22. The method of Claim 21 wherein:
2 the system includes one or more database clients;
3 said database clients are executing one or more database applications that
4 comprise said database application layer; and
5 said integrated management module is further configured to manage said one
6 or more database applications.

1 23. The method of Claim 21 wherein:
2 at least one hardware subsystem is used to execute said one or more database
3 servers; and
4 said integrated management module is further configured to manage said at
5 least one hardware subsystem.

1 24. The method of Claim 23 wherein said at least one hardware subsystem
2 includes a RAID subsystem managed by said integrated management module.

1 25. The method of Claim 21 wherein:
2 said one or more database servers reside on a local network; and
3 said console does not belong to said local network.

1 26. The method of Claim 25 wherein the console is connected to a wide area
2 network and said local network is connected to said wide area network,

3 wherein said console manages said one or more database servers through
4 messages communicated through said wide area network.

1 27. The method of Claim 25 wherein the console manages said one or more
2 database servers by messages sent over a dial up connection established
3 between said console and said local network.

1 28. The method of Claim 26 wherein said console manages said one or more
2 database servers and said one or more operating systems by changing
3 metadata stored in a centralized repository that resides on a device connected
4 to said wide area network, wherein said metadata includes data that reflects
5 configuration parameters of said one or more database servers and said one or
6 more operating systems, wherein changes made to said metadata in said
7 centralized repository are propagated over said wide area network to devices
8 on said local network.

1/10

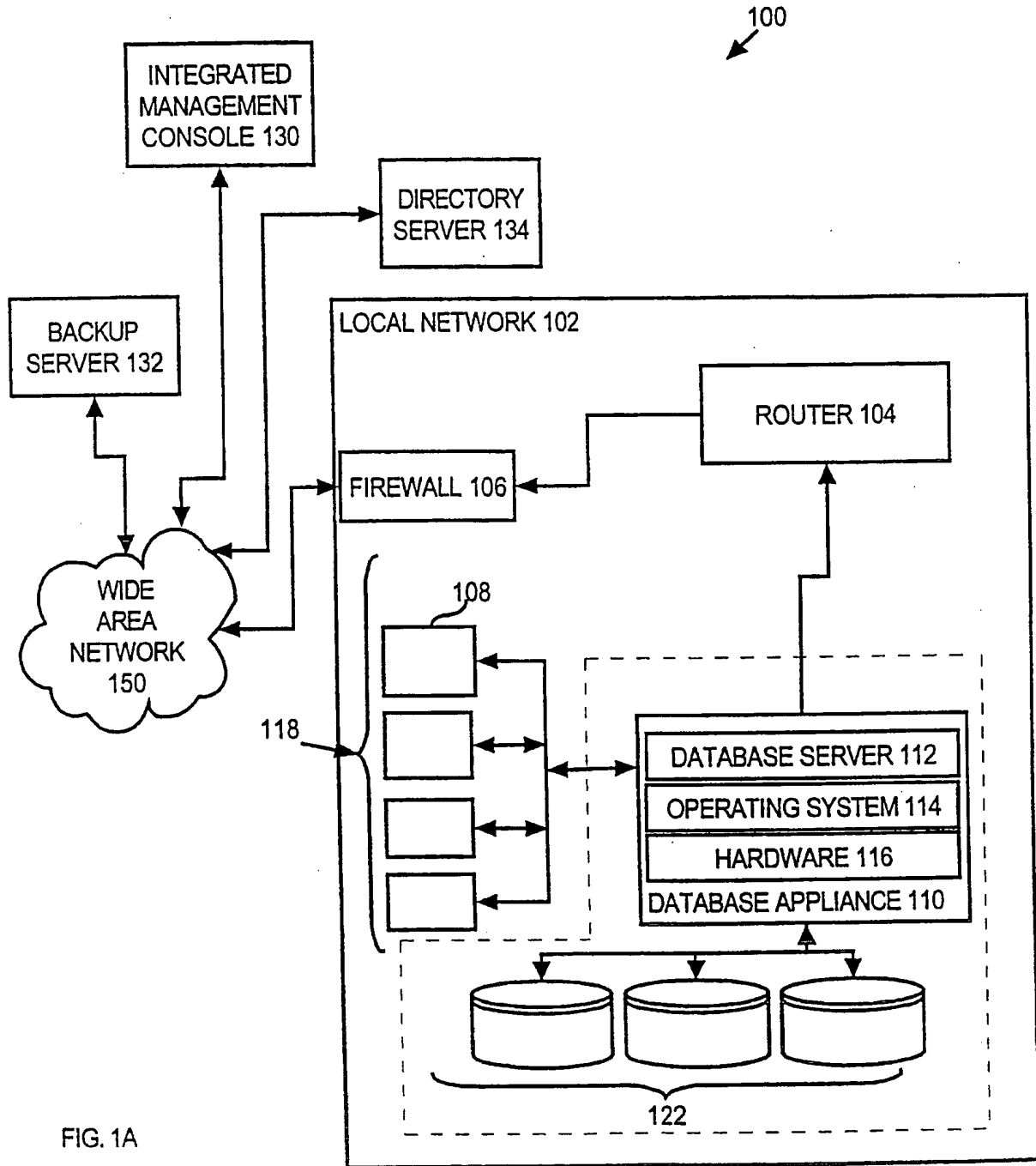
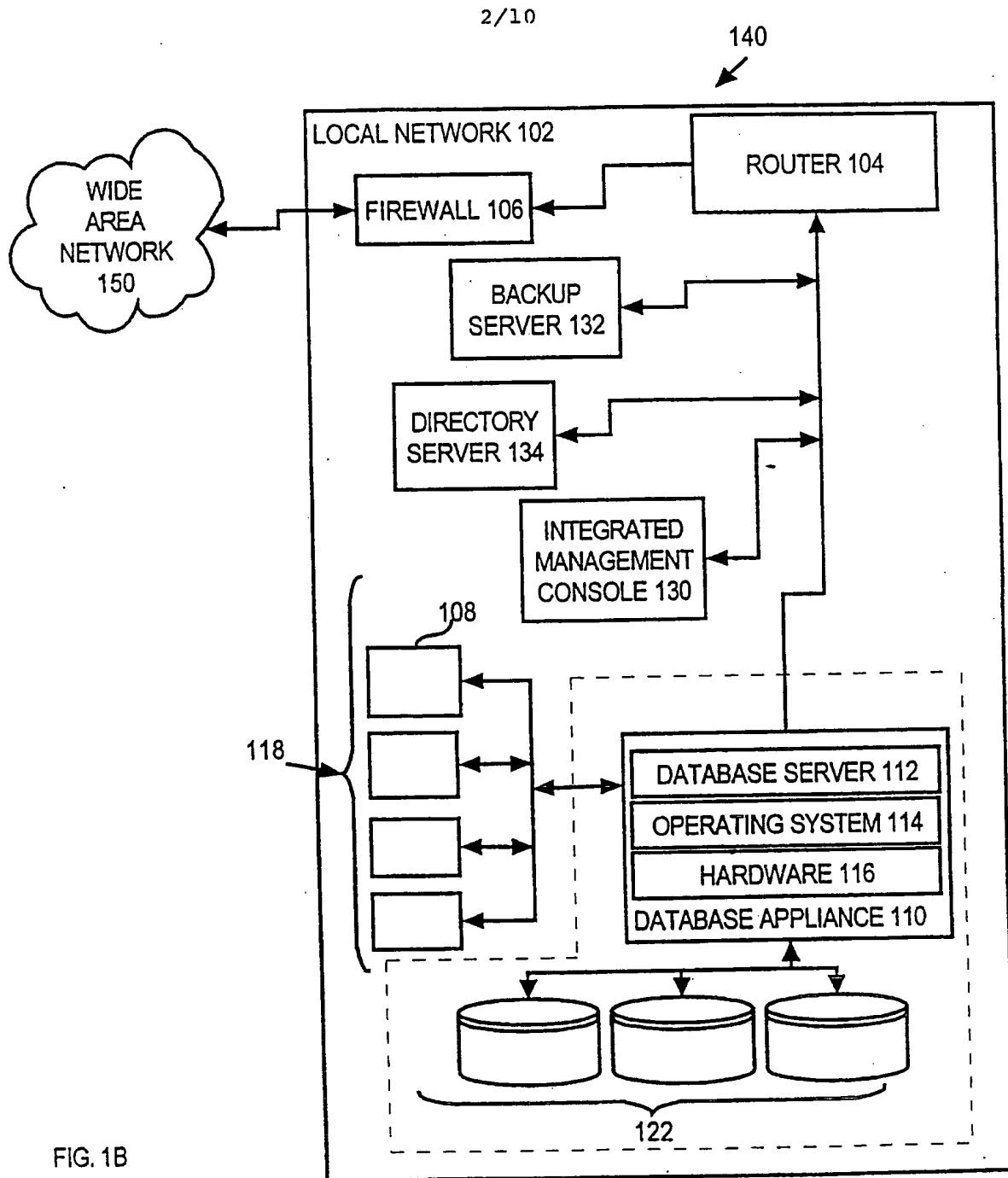


FIG. 1A



3/10

142

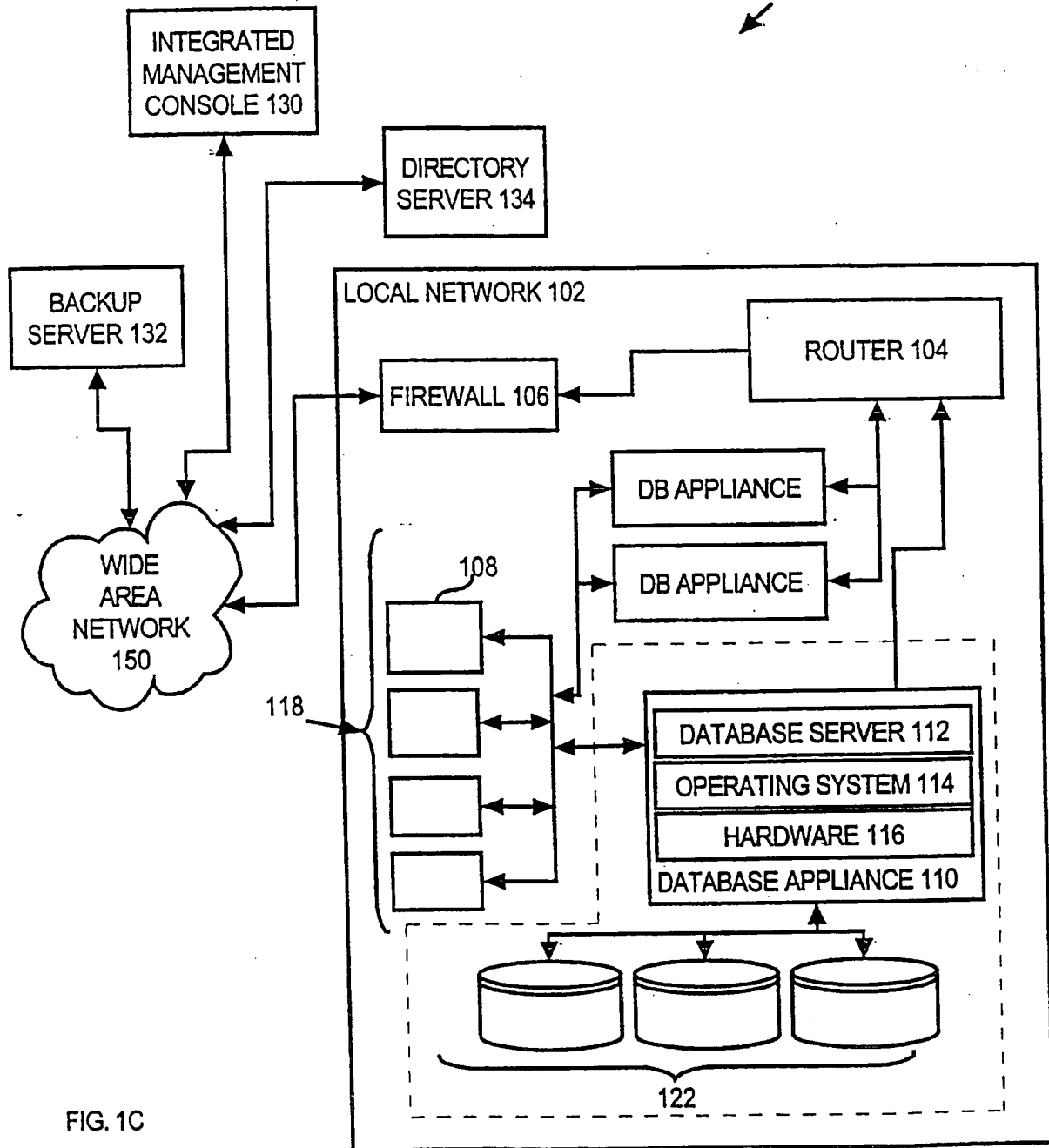


FIG. 1C

4/10

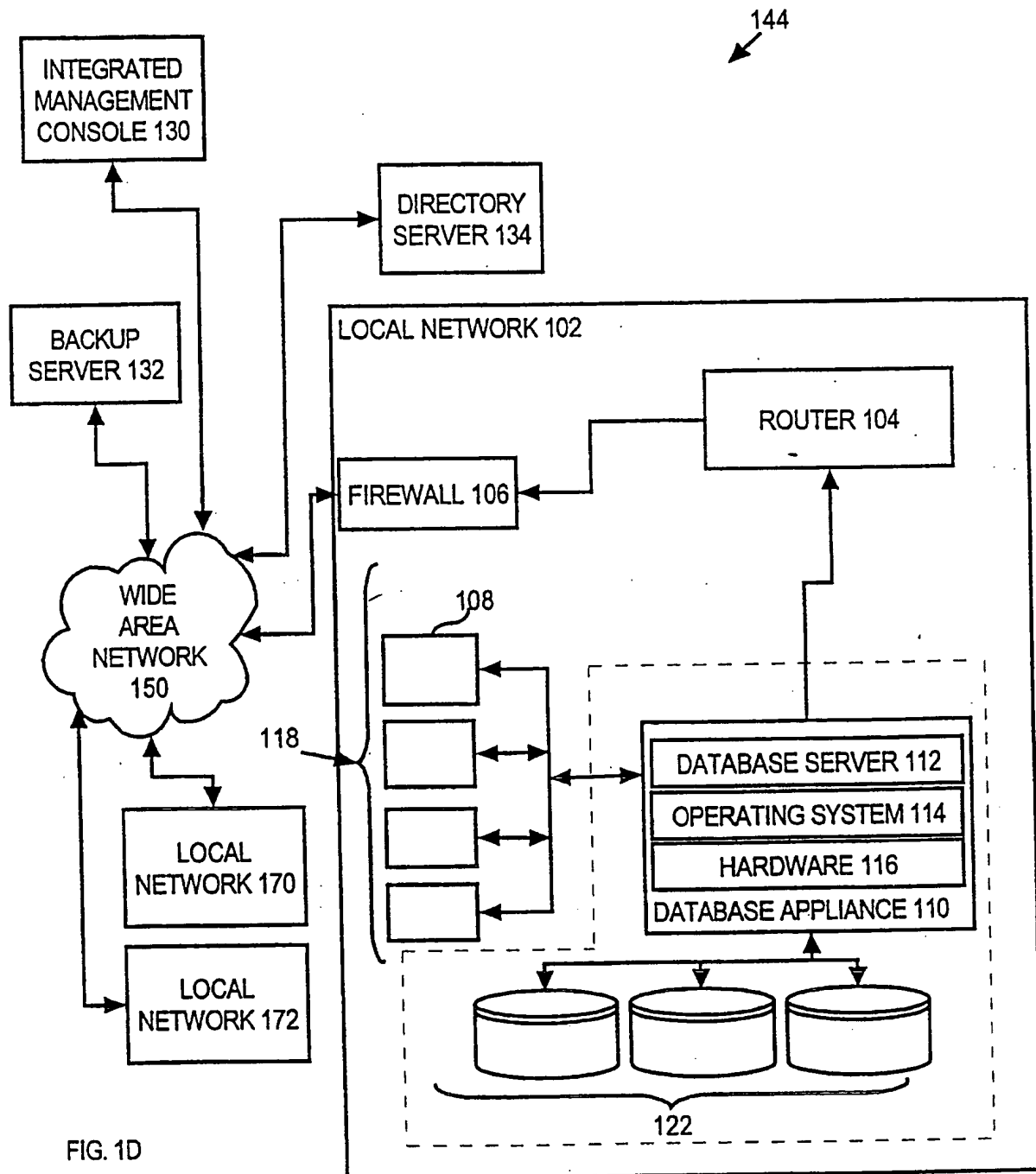
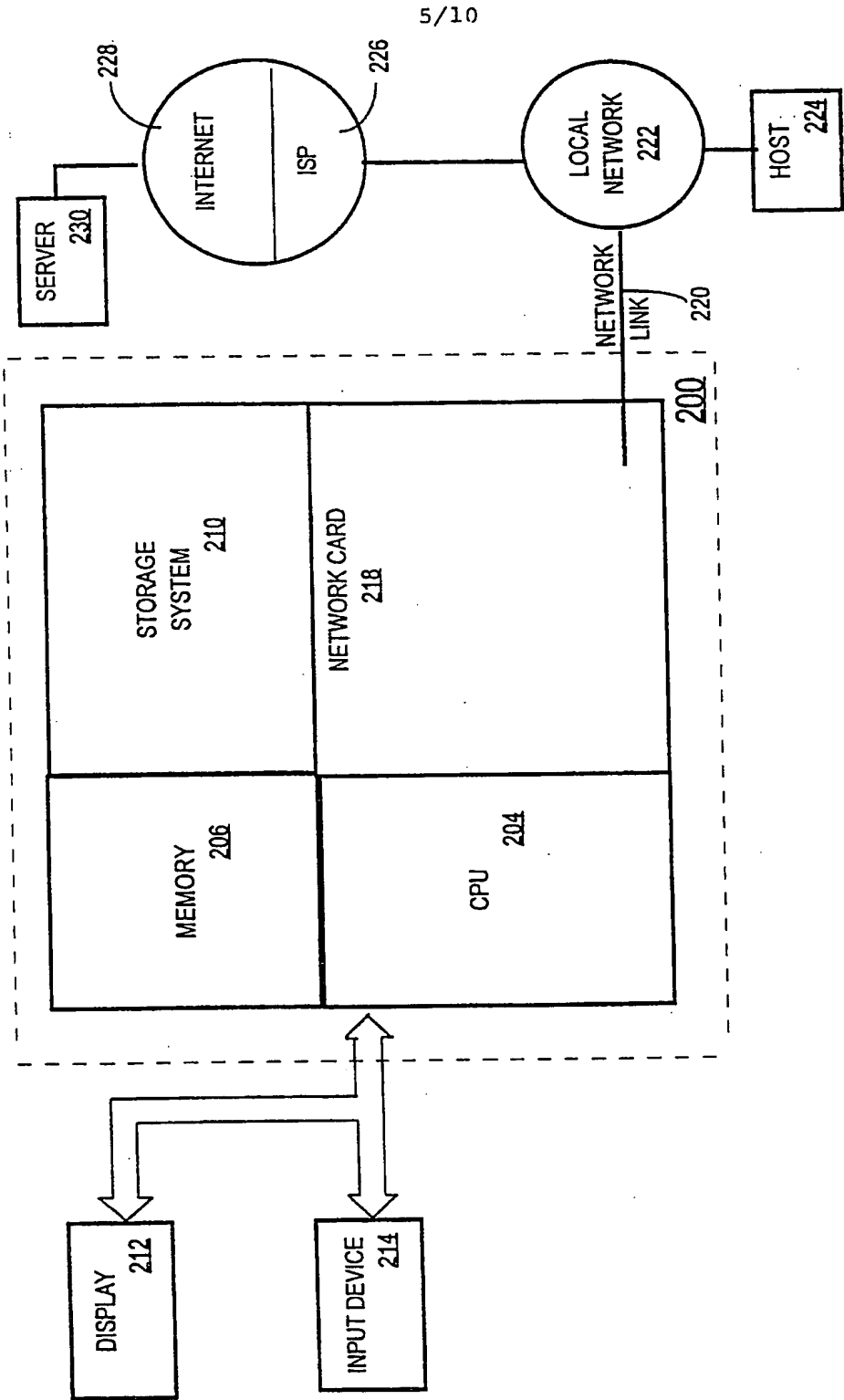
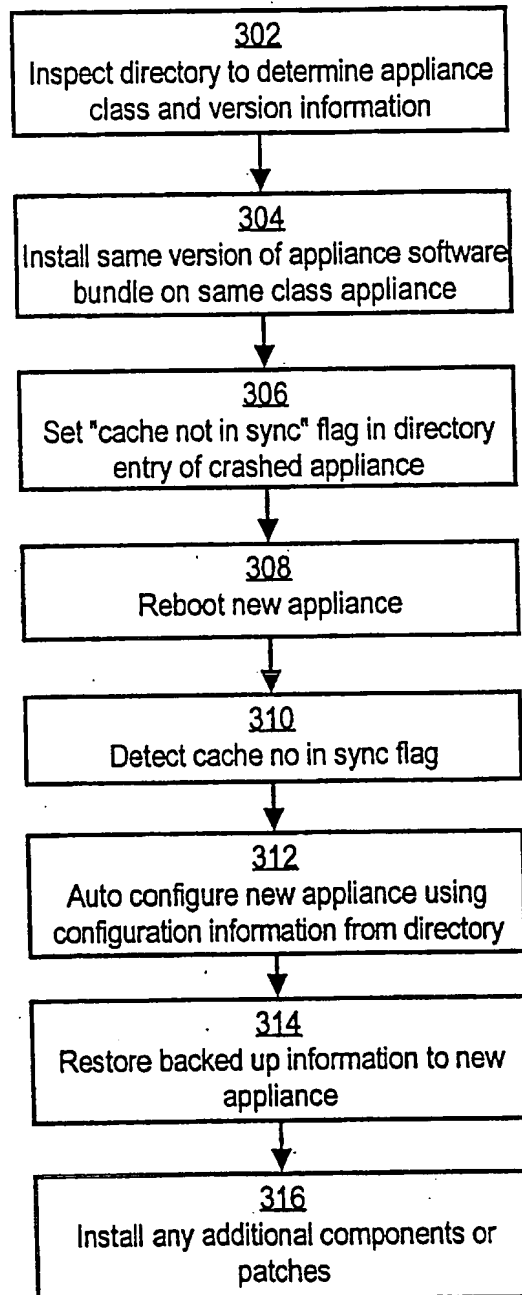


FIG. 2



6/10

**Fig. 3**

7/10

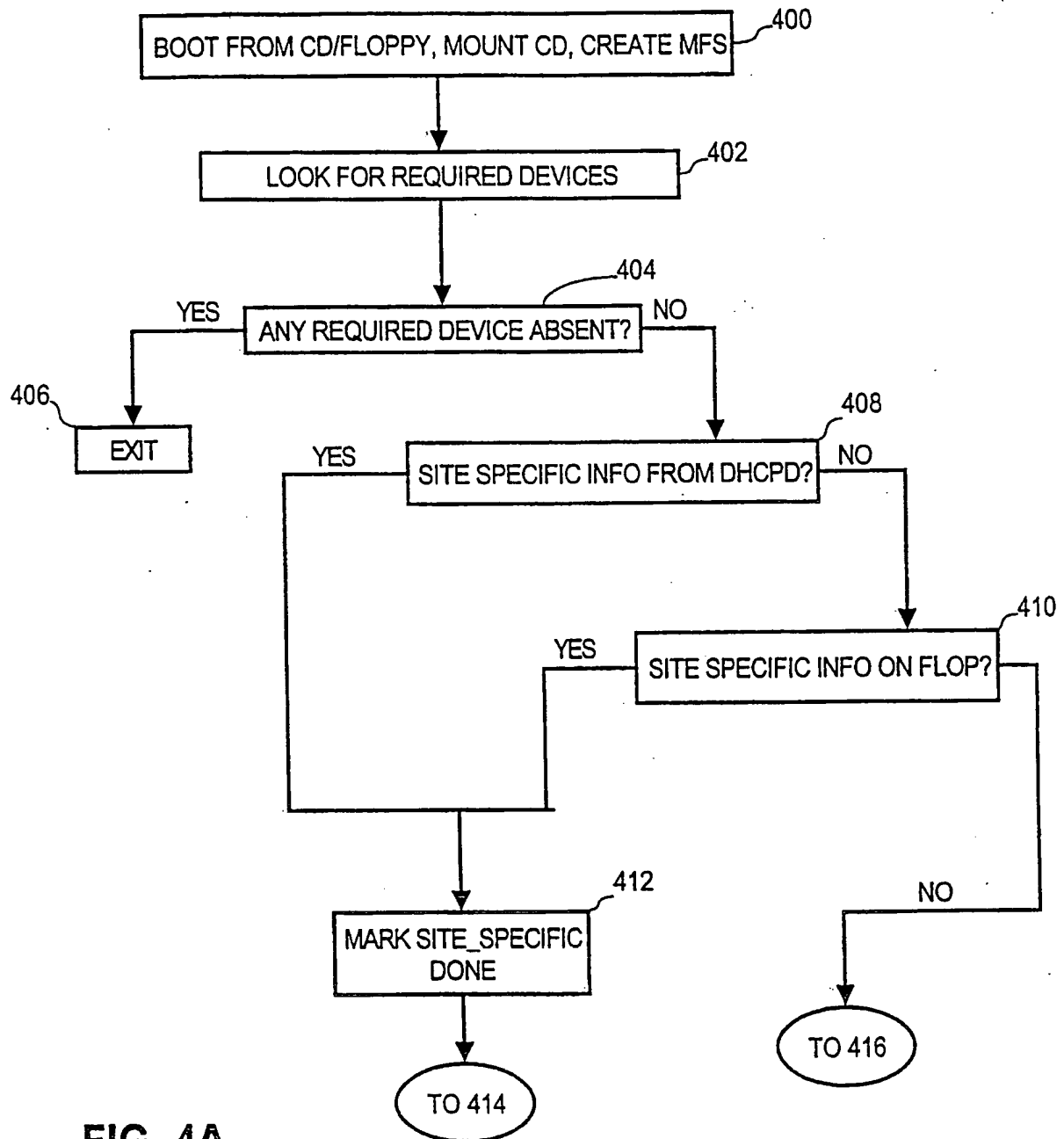
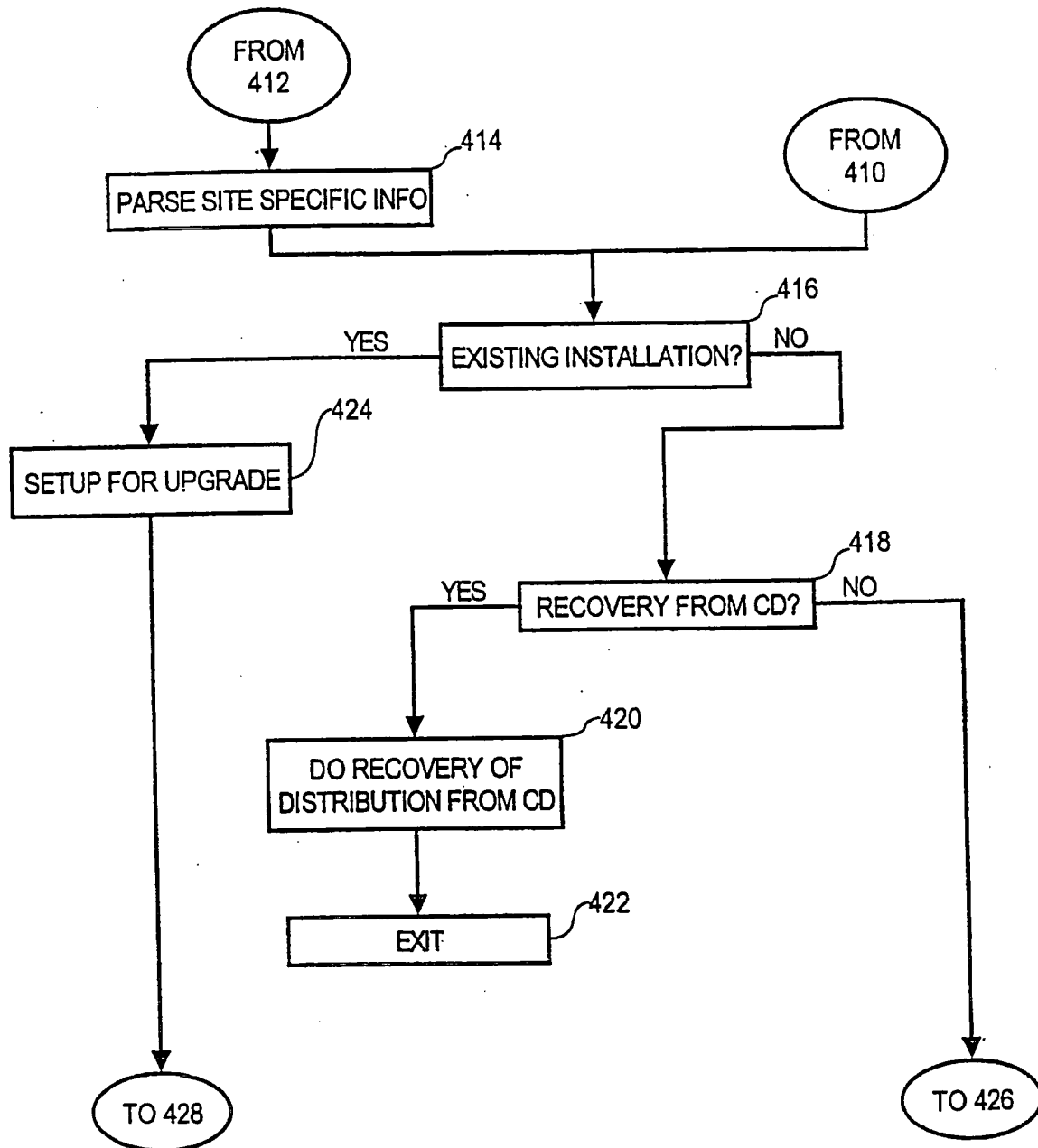


FIG. 4A

8/10

FIG. 4B



9/10

FIG. 4C

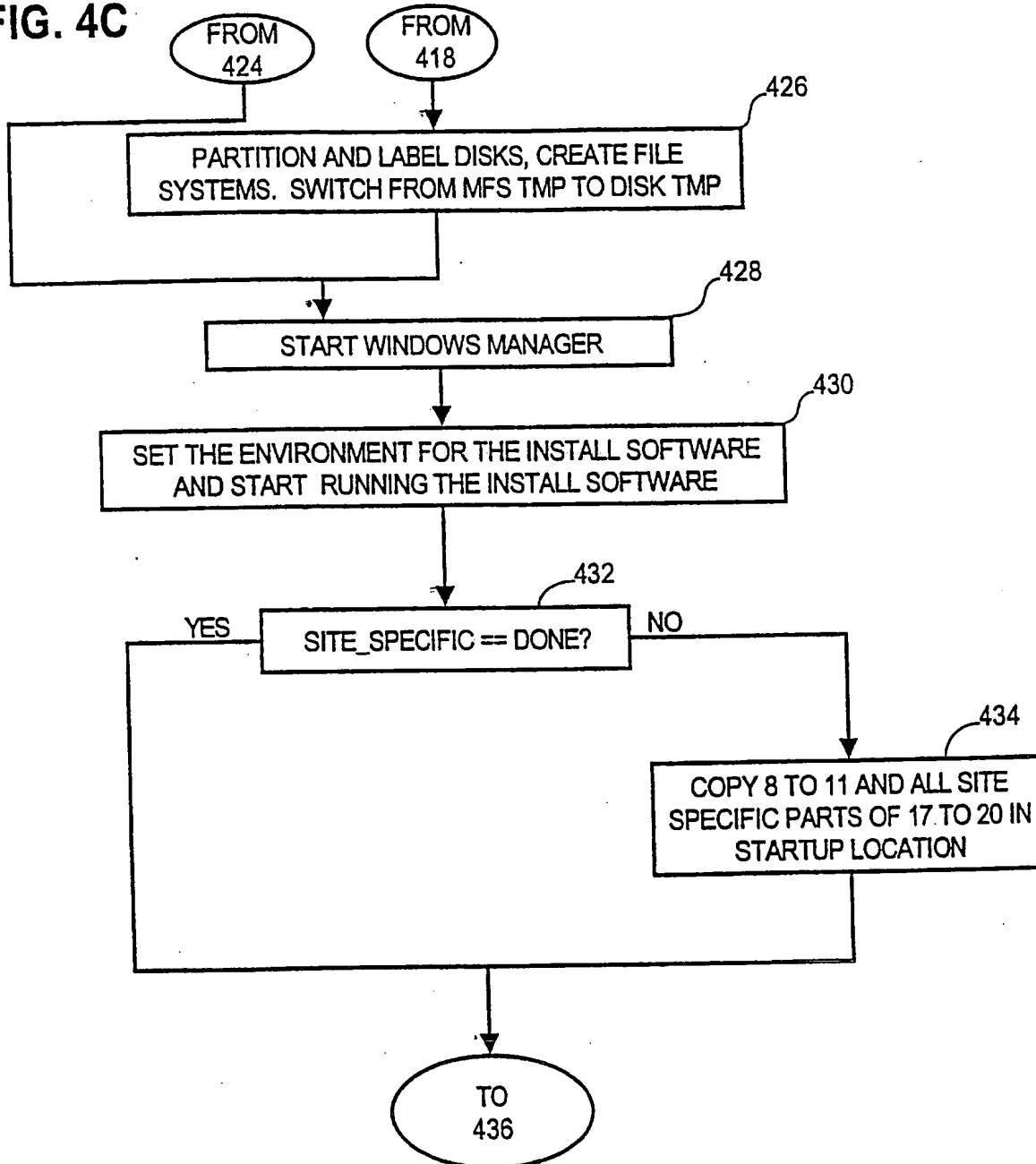
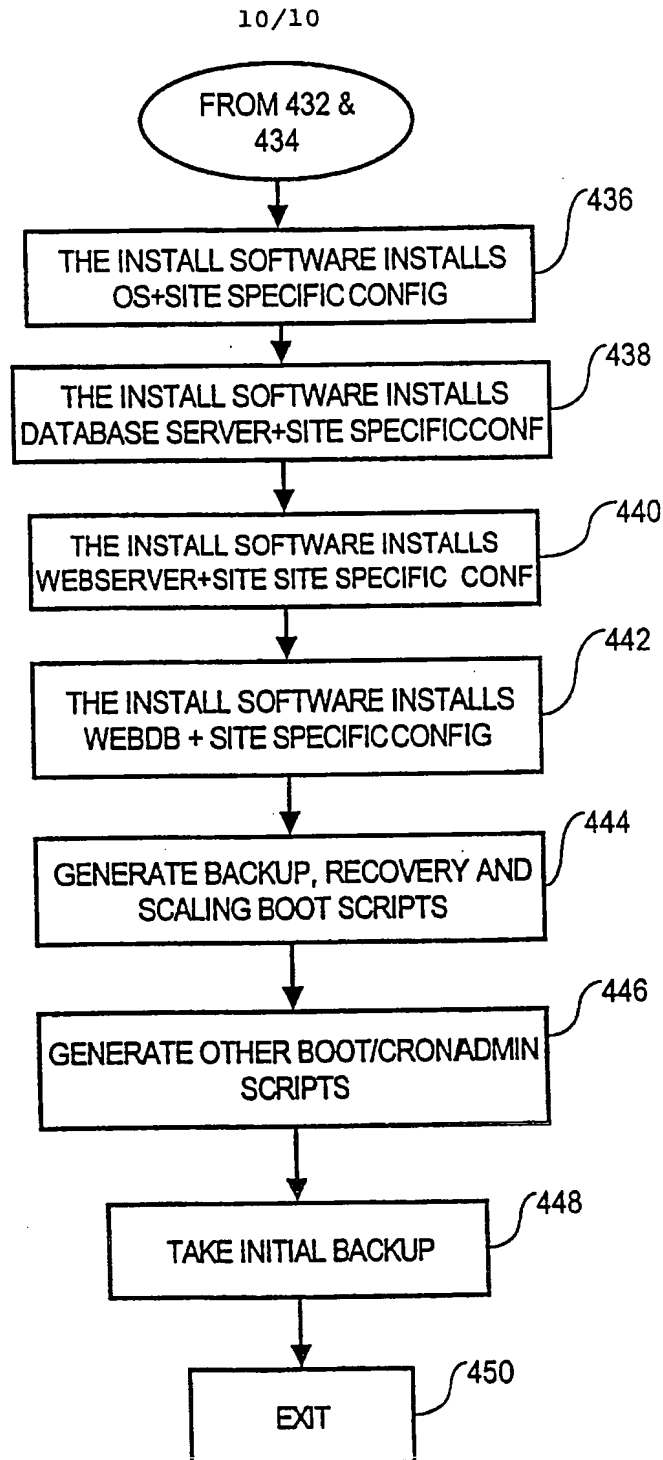


FIG. 4D



INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 00/01995

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/445 G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 274 406 A (COMPUTER X INC) 13 July 1988 (1988-07-13) abstract page 4, line 6 - line 20 page 9, line 37 - line 60 ----	1,2,4,5, 9,11
X	US 5 668 986 A (GARCIA DAVID ET AL) 16 September 1997 (1997-09-16) abstract; figure 2 column 2, line 13 - line 33 ----	15,20,21
A	EP 0 803 808 A (SUN MICROSYSTEMS INC) 29 October 1997 (1997-10-29) abstract -----	15,20

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

14 June 2000

Date of mailing of the international search report

21/06/2000

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Deane, E

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/01995

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0274406 A	13-07-1988	US 5165018 A CA 1292323 A DE 3855166 D DE 3855166 T JP 63174135 A	17-11-1992 19-11-1991 09-05-1996 14-11-1996 18-07-1988
US 5668986 A	16-09-1997	US 5606693 A	25-02-1997
EP 0803808 A	29-10-1997	US 5734831 A	31-03-1998